How to Create Cross-Platform Mobile Apps Using Xamarin.Native on Windows

Xamarin.Native is a powerful framework that allows developers to create cross-platform mobile applications with native performance and look-and-feel. It is particularly useful for developers who want to share code between iOS, Android, and Windows applications while still leveraging the native capabilities of each platform. In the Windows environment, Xamarin.Native can be integrated with Visual Studio, making it a robust solution for developers who are already familiar with Microsoft's development ecosystem.

Using Xamarin.Native on Windows, you can write your business logic in C# and share it across different platforms, while still writing native UI code for each platform. This approach provides the best of both worlds: code reuse and native performance.

Examples:

1. **Setting Up Xamarin.Native in Visual Studio on Windows:**
    - **Step 1: Install Visual Studio** Download and install Visual Studio from the official Microsoft website. During the installation, make sure to select the "Mobile development with .NET" workload.
    - **Step 2: Create a New Xamarin Project** Open Visual Studio and select "Create a new project". Choose the "Mobile App (Xamarin.Forms)" template and click "Next". Configure your project name and location, then click "Create".
    - **Step 3: Select Xamarin.Native Template** In the "New Mobile App" dialog, select the "Blank" template under "Native App" and click "Create".
2. **Writing Shared Code:**
    - **Shared Code Example:** Create a new class in the shared project to handle business logic. For example, you can create a Calculator class:

```
namespace MyXamarinApp
{
 public class Calculator
 {
     public int Add(int a, int b)
     {
         return a + b;
     }
 }
}
```

3. **Platform-Specific Code:**

   - **Android Example:** In the Android project, modify the MainActivity.cs to use the shared Calculator class:

```
using Android.App;
using Android.OS;
using Android.Widget;
using MyXamarinApp;

namespace MyXamarinApp.Droid
{
 [Activity(Label = "MyXamarinApp", MainLauncher = true)]
 public class MainActivity : Activity
 {
     protected override void OnCreate(Bundle savedInstanceState
)
     {
         base.OnCreate(savedInstanceState);
         SetContentView(Resource.Layout.activity_main);

         var calculator = new Calculator();
         int result = calculator.Add(2, 3);

         var textView = FindViewById<TextView>(Resource.Id.text
View);
         textView.Text = $"2 + 3 = {result}";
     }
 }
}
```

   - **iOS Example:** In the iOS project, modify the ViewController.cs to use the shared Calculator class:

```
using Foundation;
using UIKit;
using MyXamarinApp;

namespace MyXamarinApp.iOS
{
 public partial class ViewController : UIViewController
 {
     protected ViewController(IntPtr handle) : base(handle)
     {
     }

     public override void ViewDidLoad()
     {
         base.ViewDidLoad();
```

```
            var calculator = new Calculator();
            int result = calculator.Add(2, 3);

            resultLabel.Text = $"2 + 3 = {result}";
        }
    }
}
```

4. **Running the Application:**

    - **Android:** Connect an Android device or start an Android emulator. Set the Android project as the startup project and click "Run".

    - **iOS:** Connect a Mac with Xcode installed for remote building. Set the iOS project as the startup project and click "Run".